REMARKS

Claims 1-43 remain pending. In the present Office Action, claims 1-35 were rejected under 35 U.S.C. § 102(e) as being anticipated by Moore, U.S. Patent No. 6,594,676 ("Moore"). Claims 36-43 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Moore in view of Baird et al., "Distributed Information Storage Architecture" ("Baird"). Applicants respectfully traverse these rejections and request reconsideration.

Claims 1-35

Applicants respectfully submit that each of claims 1-35 recites a combination of features not taught or suggested in the cited art. For example, claim 1 recites a combination of features including: "said block manager is configured to change said second inode in response to updates to said first file, and wherein said block manager is configured to atomically update said first file in response to a commit of said first file by writing said second inode to said non-volatile memory".

Applicants respectfully submit that the arguments presented in the response filed July 2, 2004 (the "Previous Response") remain valid, and incorporate those arguments below to preserve them for appeal. Applicants also respond to the Response to Arguments section of the present Office Action below.

The rejection of claim 1 in the Office Action alleges that Moore teaches the above highlighted features of claim 1 at col. 2, lines 52-60. However, these teachings are: "One method for implementing database updates and commit point processing is for the database manager to maintain the database changes in storage and not apply the changes to the databases until the commit point is reached. A copy of the database data that is changed is written to the log as the update is created. When the commit point is reached, and everything went as expected, the updates are written to the databases. If something went wrong, the storage containing the database updates is freed." These teachings generally describe accumulating database updates in a log in storage, and the writing the updates to the databases when the commit point is reached. However, these teachings fail

to teach the above highlighted claim features. For example, the above teachings include no discussion of inodes, nor of performing updates atomically by writing an inode to non-volatile memory. The log is not an inode nor is the updated database data an inode. Nothing about accumulating the updates in storage, writing the updates to the database at the commit point, and using a log file to accumulate the updates teaches an inode. Furthermore, nothing in the above discussion would make such features inherent. For example, the update log may simply be a list of updates and locations in the database at which the updates are to be stored. Committing the updates may include reading the updates from the log and writing the update to the database. For example, see Moore col. 1, lines 36-42 where the log is a separate file from the database that comprises sequential records. Accordingly, Moore fails to anticipate at least "said block manager is configured to atomically update said first file in response to a commit of said first file by writing said second inode to said non-volatile memory" as recited in claim 1.

The Response to Arguments section of the present Office Action, with regard to claim 1, cites Moore, col. 1, lines 36-41; col. 2, lines 10-27; col. 2, lines 30-37; col. 10, lines 60-67 and col. 11, lines 1-3; and col. 7, lines 47-50. Applicants highlight the teachings from each of these sections individually below, to illustrate why they do not teach or suggest the combination of features recited in claim 1. Generally, these sections all have to do with databases that record database updates in a log, and use the log (sequentially, one record at a time) to restore databases after a failure. Applicants respectfully submit that nothing in these sections, either alone or taken as a whole, teaches or suggests inodes, nor "said block manager is configured to atomically update said first file in response to a commit of said first file by writing said second inode to said non-volatile memory" as recited in claim 1.

As highlighted in the Previous Response, the standard for anticipation of a claim by a reference is that the reference must teach EACH AND EVERY FEATURE of the claim (see, e.g., MPEP 2131). While the Response to Arguments section cites various portions of Moore to allegedly anticipate the above highlighted features, nothing in these sections even remotely resembles the above highlighted features. The Office Action

makes no attempt to explain why the teachings of Moore, which appear to bear little resemblance to the claimed features, are believed to anticipate such features. Rather, the Office Action simply cites numerous sections of Moore without additional comment.

At col. 1, lines 36-41, Moore teaches "As users update the database data sets in the database, the database management system records the updates into a log data set. The log data set is an amount of data, such as a file, which reflects a series of updates to the database. Log data sets are recorded in sequential records which have defined open and close points." Nothing in this section teaches or suggests inodes, nor "said block manager is configured to atomically update said first file in response to a commit of said first file by writing said second inode to said non-volatile memory".

At col. 2, lines 10-27, Moore teaches "In order to create the CADS, the change accumulation utility reads log data sets sequentially, that is, one after another. Typically, users organize their multiple databases into change accumulation groups so that the change accumulation utility operates as efficiently as possible. A user can run the change accumulation process against one change accumulation group and use an optional secondary output--the set of log records that were not written to the change accumulation data set--as input to the change accumulation utility for the next change accumulation group to be processed. This can be done for each change accumulation group in which the current change accumulation run uses the secondary output of the previous change accumulation run. This serial process is managed directly by the user. Users usually run change accumulation periodically so that when a database data set in a change accumulation group requires recovery, the time required to run a final change accumulation job and subsequent database recovery job is minimized. As can be expected, this sequential recovery process is quite complex." This section describes a sequential, one after another processing of log data sets and a sequential recovery process using change accumulation data sets and groups. Again, nothing in this section teaches inodes, nor "said block manager is configured to atomically update said first file in response to a commit of said first file by writing said second inode to said non-volatile memory".

At col. 2, lines 30-37, Moore teaches: "The recovery utility reads the entire CADS into memory and applies that portion of the CADS that is relevant to the database being restored. Each record has an identification that's sequential and the database data sets are restored in a sequential order. The recovery utility addresses each record in the CADS to see if there is a change in data for that record. If so, the CADS is accessed and the relevant record merged into the new database." This section describes sequentially, in order, recovery of database data sets. Again, nothing in this section teaches inodes, nor "said block manager is configured to atomically update said first file in response to a commit of said first file by writing said second inode to said non-volatile memory".

At col. 10, line 60-col. 11, line 3, Moore teaches: "Referring to FIG. 5 a sequence of method steps 500 is shown to illustrate one embodiment of method of the present invention. Prior to initiation of this method one or more databases 206 have failed. The recovery method initiates in step 502. Initiation may include preparing the database recovery utility for operation, for example, by creating a separate address space to manage backup data sets, CADSs, and log data sets, performing internal system checks, initializing memory and devices of required addresses, etc. Commands for implementing recovery may be executed by the database recovery utility 300 shown in FIG. 3." This section merely refers to preparing to recover databases after they have failed. Again, nothing in this section teaches inodes, nor "said block manager is configured to atomically update said first file in response to a commit of said first file by writing said second inode to said non-volatile memory".

At col. 7, lines 47-50, Moore teaches: "The database system 200 further includes one or more databases 206 having one or more database data sets. The databases 206 are designated as DB1 to DBn to illustrate a variance in the number of databases 206 in a system 200." Again, nothing in this section teaches inodes, nor "said block manager is configured to atomically update said first file in response to a commit of said first file by writing said second inode to said non-volatile memory".

Furthermore, Applicants have clarified claim 1, reciting: "a non-volatile memory storing a first inode locating a first file in said storage; and ... writing said second inode to said non-volatile memory, whereby said second inode locates said first file in said storage". Moore does not teach or suggest the above highlighted features, either.

For at least these reasons, Applicants submit that Moore fails to anticipate claim 1. Claims 2-7 depend from claim 1, and thus are not anticipated by Moore for at least the above stated reasons as well. Each of claims 2-7 recite additional combinations of features not taught or suggested in the cited art. Given that claim 1 is not anticipated by Moore, as highlighted above, additional comments regarding such additional combinations are not necessary at this time. **However, Applicants reserve the right to provide such comments on appeal.** Claim 21 is rejected as anticipated by the above discussion of Moore, and includes similar features to those highlighted above with respect to claim 1. For at least the above stated reasons, Applicants submit that Moore fails to anticipate claim 21. Claims 22-27 depend from claim 21, and thus are not anticipated by Moore for at least the above stated reasons as well. Each of claims 22-27 recite additional combinations of features not taught or suggested in the cited art. Given that claim 21 is not anticipated by Moore, as highlighted above, additional comments regarding such additional combinations are not necessary at this time. **However, Applicants reserve the right to provide such comments on appeal.**

Claim 8 recites a combination of features including: "storage is configured to copy a first inode locating said file in said storage to a second inode and to update pointers within said second inode pointing to said one or more blocks to point to said copied one or more blocks, and wherein said storage is configured to atomically update said file by writing said second inode responsive to said commit command". The rejection of claim 8 in the Office Action alleges that the above features are taught in Moore at col. 1, lines 57-65. However, these teachings are: "Database management systems include a recovery facility to respond to a database failure. Upon database failure, the recovery facility creates a new database and writes the backup copy to the new database. The recovery utility further applies all the updates to the database from

when the backup copy was created. Information used to restore the new database from the last state of the backup copy may be taken from the log data sets and recovery control information." These teachings including no discussion of inodes, updating pointers in inodes to point to copied blocks, atomic updates by writing an inode to a storage, etc. Furthermore, none of these features would be inherent in the above teachings from Moore.

The Response to Arguments section, with regard to claim 8, cites Moore, col. 1, lines 36-41; col. 7, lines 56-65; and col. 11, lines 8-23. Nothing in these sections teaches or suggests the above highlighted features of claim 8, either. These sections generally relate to using a log to accumulate updates to a database, and updating the database when a commit point is reached. Additionally, these teachings refer to recovering the database from a failure using the logs. Nothing in these teachings teaches or suggests "storage is configured to copy a first inode locating said file in said storage to a second inode and to update pointers within said second inode pointing to said one or more blocks to point to said copied one or more blocks, and wherein said storage is configured to atomically update said file by writing said second inode responsive to said commit command" as recited in claim 8.

At col. 1, lines 36-41, Moore teaches: "As users update the database data sets in the database, the database management system records the updates into a log data set. The log data set is an amount of data, such as a file, which reflects a series of updates to the database. Log data sets are recorded in sequential records which have defined open and close points." The log data sets are sequential records, each record comprising an update to the database. Nothing in this section teaches or suggests "storage is configured to copy a first inode locating said file in said storage to a second inode and to update pointers within said second inode pointing to said one or more blocks to point to said copied one or more blocks, and wherein said storage is configured to atomically update said file by writing said second inode responsive to said commit command" as recited in claim 8.

At col. 7, lines 56-65, Moore teaches: "Each database management system 202 may include a log 204 having log records to track updates to data kept in memory 18 or in a database 206. The log 204 is used for reference to track data changes and other events performed by the corresponding database management system 202. Changes and other events are stored on the log 204 as log records. The log 204 may be stored on one or more memory devices 18 of the station 12." Again, the log records are the database updates. Nothing in this section teaches or suggests "storage is configured to copy a first inode locating said file in said storage to a second inode and to update pointers within said second inode pointing to said one or more blocks to point to said copied one or more blocks, and wherein said storage is configured to atomically update said file by writing said second inode responsive to said commit command" as recited in claim 8.

At col. 11, lines 8-23, Moore teaches: "In step 504, the recovery utility 300 builds a recovery list which is a collection of databases 206 to be recovered. In one embodiment, when a recovery list is built in step 504, it is associated with a logical terminal that issued the recovery command. Recovery continues in step 506 when the recovery utility 102 receives a command to start the recovery. The recovery utility 300 performs a check to determine if recovery is currently in process or if a desired recovery list cannot be found. If so, an error message issues and recovery is aborted. Otherwise, recovery continues. The recovery utility 300 validates the recovery list by ensuring that each database 206 is in a state that allows it to be recovered, and also determines the resources needed for recovery of these validated entries." Nothing in this section has anything to do with "storage is configured to copy a first inode locating said file in said storage to a second inode and to update pointers within said second inode pointing to said one or more blocks to point to said copied one or more blocks, and wherein said storage is configured to atomically update said file by writing said second inode responsive to said commit command" as recited in claim 8.

Furthermore, claim 8 recites a combination of features including: "said first inode is stored in an inode file, and wherein said inode file is identified by a master inode, and wherein said inode file is atomically updated with said second inode by writing said

master inode subsequent to said commit command". The same teachings highlighted above are also alleged to teach these features of claim 8. Nothing in the above teachings has anything even remotely to do with the above highlighted features of claim 8.

Accordingly, Moore fails to anticipate claim 8. Claims 9-10 depend from claim 1, and thus are not anticipated by Moore for at least the above stated reasons as well. Each of claims 9-10 recite additional combinations of features not taught or suggested in the cited art. Given that claim 8 is not anticipated by Moore, as highlighted above, additional comments regarding such additional combinations are not necessary at this time. **However, Applicants reserve the right to provide such comments on appeal.**

Claim 11 recites a combination of features including: "modifying said second inode in response to one or more changes to said first file; and atomically updating said first file by establishing said second inode as the inode for said first file". The Office Action alleges that these feature are taught in Moore at col. 2, lines 12-29. However, these teachings are: "Typically, users organize their multiple databases into change accumulation groups so that the change accumulation utility operates as efficiently as possible. A user can run the change accumulation process against one change accumulation group and use an optional secondary output--the set of log records that were not written to the change accumulation data set--as input to the change accumulation utility for the next change accumulation group to be processed. This can be done for each change accumulation group in which the current change accumulation run uses the secondary output of the previous change accumulation run. This serial process is managed directly by the user. Users usually run change accumulation periodically so that when a database data set in a change accumulation group requires recovery, the time required to run a final change accumulation job and subsequent database recovery job is minimized. As can be expected, this sequential recovery process is quite complex." These teachings include no discussion of, e.g., inodes and atomic updates. Furthermore, these features would not be inherent in the above discussion.

Furthermore, Applicants have amended claim 11 to recite: "said first inode

locates a first file in a storage". Moore does not teach or suggest the above highlighted features, in combination with the other features of claim 11.

Accordingly, Moore fails to anticipate claim 11. Claims 12-20 depend from claim 11, and thus are not anticipated by Moore for at least the above stated reasons as well. Each of claims 12-20 recite additional combinations of features not taught or suggested in the cited art. Given that claim 11 is not anticipated by Moore, as highlighted above, additional comments regarding such additional combinations are not necessary at this time. **However, Applicants reserve the right to provide such comments on appeal.** Claim 28 is rejected as anticipated by the above discussion of Moore, and includes similar features to those highlighted above with respect to claim 11. For at least the above stated reasons, Applicants submit that Moore fails to anticipate claim 28. Claims 29-35 depend from claim 28, and thus are not anticipated by Moore for at least the above stated reasons as well. Each of claims 29-35 recite additional combinations of features not taught or suggested in the cited art. Given that claim 28 is not anticipated by Moore, as highlighted above, additional comments regarding such additional combinations are not necessary at this time. **However, Applicants reserve the right to provide such comments on appeal.**

The Office Action alleges that claims 4, 9, 19, and 24 and claims 5, 10, 20, and 25 are inherent. Applicants respectfully disagree. Furthermore, Applicants note that the explanation of why these claims are allegedly inherent appears to quote Applicants own disclosure. In the case of claims 4, 9, 19, and 24, the Office Action appears to quote the specification page 8, lines 22-24. In the case of claims 5, 10, 20, and 25, the Office Action quotes the specification, page 14, lines 17-19. While it is permissible to use a secondary reference to show inherency of a feature (MPEP 2131.01(III)), that secondary reference must be prior art. It is certainly improper to use Applicants' own specification to attempt to prove inherency. Furthermore, for a feature to be inherent, the missing description must necessarily be present in the thing described in the reference (again, see MPEP 2131.01(III)). Applicants submit that the Office Action has not demonstrated that

the features of claims 4-5, 9-10, 19-20, and 24-25 are necessarily present, and Applicants submit that the features are not inherent.

In the Response to Arguments section, the present Office Action refers to the commands for implementing recovery (Moore, col. 11, lines 1-3) and further refers to the teachings at col. 10, lines 12-26. Applicants respectfully submit that commands for implementing recovery have nothing to do with commit commands. Furthermore the teachings at col. 10, lines 12-26 have no description of any commands, but merely describe part of the database recovery utility.

Claims 36-43

Applicants respectfully submit that each of claims 36-43 recite combinations of features not taught or suggested in the cited art. For example, claim 36 recites a combination of features including: "the block manager is configured to atomically update the first file to reflect the plurality of write commands responsive to the commit command".

The alleged combination of Moore and Baird fails to form a *prima facie* case of obviousness because the alleged combination does not teach or suggest all of the claimed features. The present Office Action states that Moore does not teach or suggest features including the above highlighted features of claim 36 (see Office Action, page 11, item 16). However, the Office alleges that Baird does teach the above highlighted features. Specifically, the Office Action states that Baird discloses "a distributed information storage architecture having an information storage management divides [sic] systems into the following major domains: information (intelligence applied to facts), data (raw un-interpreted facts), and storage (repository for data). For example, applications use files; files use volumes and volumes use devices. In this example, an application (in the information domain) stores information in files. A file manager (in the data domain) stores files on volumes. A volume manager (in the storage domain) keeps track of volumes and mounts volumes on devices (also in the storage domain) and a device driver schedules I/O requests" (Office Action, page 11, item 16, lines 8-15).

Nothing in this citation is even remotely related to "the block manager is configured to atomically update the first file to reflect the plurality of write commands responsive to the commit command".

The alleged combination of Moore and Baird fails to form a *prima facie* case of obviousness because the a proper motivation to combine has not been established. The Office Action concludes that it would be obvious to employ the distributed information storage architecture disclosed by Baird to the recovery utility apparatus disclosed by Moore because it would be an advancement to provide a simplified database recovery apparatus and systems that substantially reduces recovery time after a database failure (Office Action, page 11, item 16, lines 15-19). However, this reasoning comes from Moore (see Moore, col. 4, lines 37-39) and is in fact the advancement that Moore accomplishes with his disclosure (see Moore, col. 4, line 49). Accordingly, Moore has no need for Baird to provide the simplified database recovery apparatus and systems that substantially reduces recovery time after a database failure. Furthermore, it is not at all clear how Baird's system would provide an simplified database recovery apparatus, or whether Baird's system even has anything to do with a database recovery apparatus. Certainly, the description above of Baird's domains, applications, file manager, and volume manager have nothing to do with as simplified database recovery apparatus.

For at least the above stated reasons, Applicants submit that claim 36 is patentable over the cited art. Claims 37-39, being dependent from claim 36, are similarly patentable over the cited art. Each of claims 37-39 recites additional combinations of features not taught or suggested in the cited art. Given that claim 36 is patentable, as highlighted above, additional comments regarding such additional combinations are not necessary at this time. **However, Applicants reserve the right to provide such comments on appeal.**

Claim 40 recites a combination of features including: "the storage is configured to atomically update the first file to reflect the plurality of write commands responsive to the commit command". The Office Action alleges that claim 40 is taught in Baird,

stating "An enterprise can configure several storage nodes (each with their own contingent of storage devices) into a single storage system. An administrator can add a storage node to a storage system or remove one without disrupting work on other storage nodes. The relationship between two storage nodes is illustrated in Figure 8" (Office Action, page 12, item 20). Applicants respectfully submit that nothing in the above teachings from Baird and nothing in Baird's Figure 8 has anything to do with "the storage is configured to atomically update the first file to reflect the plurality of write commands responsive to the commit command". Thus, the alleged combination of Moore and Baird also does not form a *prima facie* case of obviousness since the combination does not teach or suggest all the features of claim 40. Furthermore, as highlighted above with regard to claim 36, the alleged combination of Moore and Baird also does not form a *prima facie* case of obviousness because a proper motivation to combine has not been established.

Accordingly, claim 40 is patentable over Moore and Baird. Claims 41-43 being dependent from claim 40, are similarly patentable over the cited art. Each of claims 41-43 recites additional combinations of features not taught or suggested in the cited art. Given that claim 40 is patentable, as highlighted above, additional comments regarding such additional combinations are not necessary at this time. **However, Applicants reserve the right to provide such comments on appeal.**

Information Disclosure Statement (IDS)

Applicants filed an Electronic IDS on July 2, 2004. Applicants have not yet received the initial PTO-1449 form included in that IDS to evidence consideration of the cited references. Applicants have attached hereto a copy of the Electronic IDS and the Acknowledgement Receipt from the PTO evidencing receipt of the IDS on July 2, 2004. Applicants respectfully request a return of the PTO-1449 form from the Electronic IDS, initialed and signed by the Examiner to evidence consideration of the cited references.

Interview Summary

On February 10, 2005, the undersigned had an interview with the Examiner in this application. Proposed claim amendments similar in nature to those made in this Response were discussed. The prior art was also discussed, including Moore and Baird. Arguments similar to those presented in this Response were discussed, highlighting reasons why the claims are patentable over the cited art.

CONCLUSION

Applicants submit that the application is in condition for allowance, and an early notice to that effect is requested.

If any extensions of time (under 37 C.F.R. § 1.136) are necessary to prevent the above referenced application(s) from becoming abandoned, Applicant(s) hereby petition for such extensions. If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5181-59100/LJM.

Also enclosed herewith are the following items:

☒ Return Receipt Postcard

☐ Petition for Extension of Time

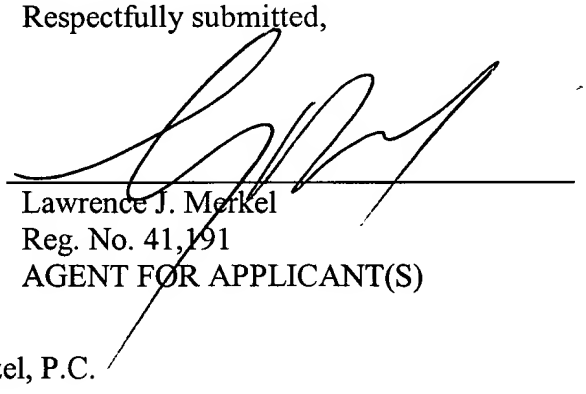
☐ Request for Approval of Drawing Changes

☐ Notice of Change of Address

☐ Please debit the above deposit account in the amount of $ for fees ( ).

☒ Other: Copy of Electronic IDS submitted July 2, 2004

Respectfully submitted,



Lawrence J. Merkel
Reg. No. 41,191
AGENT FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8800

Date: 2/10/05